

Abstract: Molecular Dynamics (MD) is an important step towards understanding the behavior of biological systems. Even though, the state-of-the-art computational resource and techniques have made it feasible to do large-scale MD simulations, MD remains a very compute-intensive job, especially for large realistic systems. Thus there exists interest in developing and understanding the performance of MD code on multiple distributed HPC resources. To realize this goal, we have developed a preliminary parallel and distributed MD code and benchmarked it on the LONI (Louisiana Optical Network Interface) grid, which uses dedicated light-path networks to connect supercomputers across Louisiana. This work is motivated by an attempt to utilize new infrastructure and to devise new programming strategies for MD simulations, with a focus on distributing the workload across various machines while retaining the advantages of parallelization within a single machine. Current practice is to distribute the job amongst various machines only when the resource requirement is more than the capacity of a single machine. In our work, we divided the job into several workloads, even if a single machine was capable of handling it. We tested our developed code on upto three distributed resources of LONI, namely Bluedawg, Zeke and Ducky. These are IBM P5 clusters with 114 nodes per cluster. Based on performance data, we show that without any serious optimization, the performance degradation as defined by total CPU-hrs on multiple machines is about 10-20% of the performance over a single machine, which has useful consequences when time to finish is critical. Based on this analysis, the users of grid resources can pick their choice between two different strategies: (1) optimize the CPU-hours used, and live with the huge wait time involved, (2) or use an extra 10-20% CPU-hours and optimize the overall job throughput. Our analysis has the potential to be extended to parallel codes other than MD. We believe that this study can lead to a better job distribution and resource allocation to optimize throughput.

Molecular Dynamics

- Atoms stretch, vibrate, and rotate about the bonds in response to intermolecular and intra-molecular forces.
- Involves both bonded and non-bonded forces
- Equations to determine velocity and displacement:

$$-(V_i)_{t+1} = (V_i)_t + (F_i / M_i)_t$$

$$-(X_i)_{t+1} = (V_i)_t + (X_i)_t$$

The parallel MD code

- Extended on the existing Mindy code for MD
- Computation of non-bonded forces is the most compute-intensive part, as seen from profiling of the code
- Non-bonded part takes 72-80% of computation time
- Bonded parts (to compute bond-angles, improper and dihedral angles) takes 3-5% of computation time

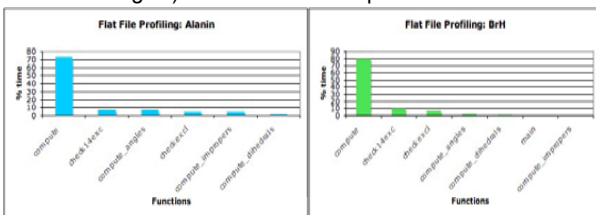


Fig: Flat-file profiling (Alanine)

Fig: Flat-file profiling (BrH)

Test-sets: Proteins

•Alanin (Alanine)

- size 66 atoms

•BrH (Bacteriorhodopsin)

- size 3762 atoms



Source: www.wiki.org



Source: www.pdb.org

Test-bed: LONI (Louisiana Optical Network Initiative)

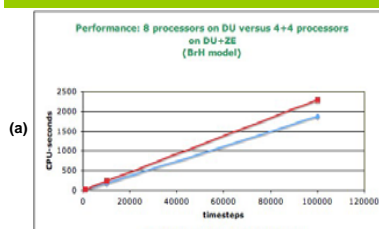
LONI is nation's one of the first network to connect Louisiana's supercomputers with fast light-paths so as to minimize communication delay.

We used LONI because, we wanted to know:

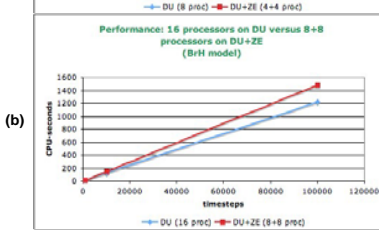
- The behavior of a distributed code, assuming minimum communication delay possible
- Exactly performance degradation a code can have, when run in distributed mode in supercomputers
- We used 3 LONI machines: Bluedawg, Ducky, Zeke
- LONI uses Loadleveler for job scheduling, which implements the backfill algorithm on FIFO scheme.



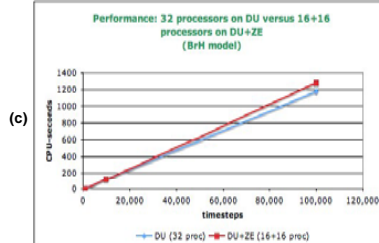
Performance Analysis



timesteps	Total CPU-hrs (sec)	
	DU (8 proc)	DU+ZE (4+4 proc)
100	1.5	1.7
1,000	19.5	23.8
10,000	191.5	234.3
100,000	1882.7	2308.1
1,000,000		28132.5

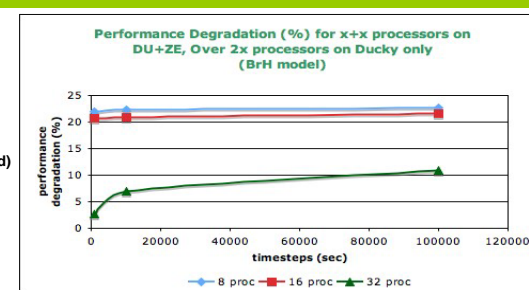


timesteps	Total CPU-hrs (sec)	
	DU (16 proc)	DU+ZE (8+8 proc)
100	0.0	0.0
1,000	11.1	13.4
10,000	123.7	149.6
100,000	1219.0	1481.9
1,000,000		57494.3



timesteps	Total CPU-hrs (sec)	
	DU (32 proc)	DU+ZE (16+16 proc)
100	0.0	0.0
1,000	3.7	3.8
10,000	116.0	124.0
100,000	1175.5	1282.7
1,000,000	11977.0	13276.9

Fig: Performance degradation when distributed across 2 LONI machines is about 10-20%; (a), (b), (c) shows actual values; (d) shows percentage-comparison; BL, DU, ZE are LONI machines, representing Bluedawg, Ducky, Zeke.



The graphs and tables shown depicts performance for Bacteriorhodopsin protein dataset. Similar performance is seen on Alanine protein dataset - a much smaller size dataset.

Conclusion

- Performance degradation of compute-intensive code like MD is 10-20% of the performance over single machine
- Hence, users can choose any one:
 - optimize CPU-hours used, and try to get maximum resource over a single machine
 - optimize job throughput, because of lesser wait time, by using extra 10-20% CPU-hours

Acknowledgements / Sponsorships

This research has been supported by:

- Center for Computation and Technology (CCT)
- Cybertools Project

